


Join the Stack Overflow Community

Stack Overflow is a community of 6.7 million programmers, just like you, helping each other.
Join them; it only takes a minute:

Sign up

Binding Sockets to IPv6 Addresses

```
36 if (dev.isBored() || job.sucks()) {
37     searchJobs({flexibleHours: true, companyCulture: 100});
38 }
39 // A career site that's by developers, for developers.
```


[Get started](#)

I am trying to write a web server that listens on both IPv4 and IPv6 addresses. However, the code that I originally wrote did not work. Then I found out that the IPv6 structures word for both IPv4 and IPv6. So now I use the IPv6 structures however, only the IPv4 addresses work. This post, [why can't i bind ipv6 socket to a linklocal address](#), which said to to add `server.sin6_scope_id = 5;` so I did that but it still does not accept IPv6 telnet connections. Any help would be greatly appreciated because I am thoroughly stumped.
Thanks!

My code is below:

```
void initialize_server(int port, int connections, char* address)
{
    struct sockaddr_in6 socket_struct;
    /*Creates the socket*/
    if ((sock_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        syslog(LOG_ERR, "%s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }/*Ends the socket creation*/

    /*Populates the socket address structure*/
    socket_struct.sin6_family = AF_INET6;

    if(address == NULL)
        socket_struct.sin6_addr=in6addr_any;
    else
    {
        inet_pton(AF_INET6, "fe80::216:3eff:fec3:3c22", (void
*)&socket_struct.sin6_addr.s6_addr);
    }
    socket_struct.sin6_port =htons(port);
    socket_struct.sin6_scope_id = 0;
    if (bind(sock_fd, (struct sockaddr*) &socket_struct, sizeof(socket_struct))
< 0)
    {
        syslog(LOG_ERR, "%s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }//Ends the binding.

    if (listen(sock_fd, connections) <0)
    {
        syslog(LOG_ERR, "%s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }//Ends the listening function


} //ends the initilaize server function.
```

c sockets ipv6 ipv4 berkeley-sockets

edited Aug 20 '16 at 18:41

 Navin
1,374 1 9 27

asked Nov 22 '12 at 2:06

 tpar44
600 4 10 30

- what i know ipv6 is 64-bit, so servers using ipv6 must be running on a 64-bit operating systems. just use ipv4 for now, there's alot of issues using ipv6 so far. – [GiantHornet](#) Nov 22 '12 at 2:09
- 2 @GiantHornet: IPv6 is neither 32-bit nor 64-bit; it can run on either and on other systems too. – [Jonathan Leffler](#) Nov 22 '12 at 2:10
- @GiantHornet yea, I don't think thats right because my ubuntu machine is i686 which is 32-bit and has a IPv6 address – [tpar44](#) Nov 22 '12 at 2:12
- 2 @GiantHornet IPv6 is 128-bit long, but I don't know how OS architecture could be important here. Also, suggesting to drop support for IPv6 is the root of the problems with IPv6. – [Crozin](#) Nov 22 '12 at 2:12
- 2 @GiantHornet well if im going to try to learn sockets, IPv4 seems to be pretty easy, i might as well try IPv6 while I'm at it – [tpar44](#) Nov 22 '12 at 2:19

2 Answers

You're creating a socket in the `AF_INET` family, but then trying to bind it to an address in the `AF_INET6` family. Switch to using `AF_INET6` in your call to `socket()`.

answered Nov 22 '12 at 3:03



qqx

10.8k 1 27 52

Thanks for the hint but it still doesn't work... – tpar44 Nov 22 '12 at 3:39

Making that change got it working for me. – qqx Nov 22 '12 at 3:40

just out of the curiosity you use the `telnet fe80::216:3eff:fec3:3c22%eth0 8080` to test it? – tpar44 Nov 22 '12 at 3:42

I passed NULL as the address to that function so that it would bind to `in6addr_any`. Then I used `telnet 0 6666` to connect to it. I'm not able to connect to link local addresses that other ports are listening on either. – qqx Nov 22 '12 at 3:49

Saying `"server.sin6_scope_id = 5;"` is arbitrary. I fought with this awhile myself and discovered you need to use the actual scope of the actual interface you want to bind on. It can be found with an obscure but useful little function.

```
#include <net/if.h>
server.sin6_scope_id=if_nametoindex("eth0");
```

Of course, hardcoding it to one particular adapter is bad, shortsighted coding. A more complete solution is to loop through all of them and match on the ip address you're binding. The following is not perfect in that it doesn't account for quirks like having non-canonical addresses and two adapters with the same ip, etc. But besoverall, this sample function works great and should get you started.

```
#include <string.h> // strcmp
#include <net/if.h> // if_nametoindex()
#include <ifaddrs.h> // getifaddrs()
#include <netdb.h> // NI_ constants

// returns 0 on error
unsigned getScopeForIp(const char *ip){
    struct ifaddrs *addrs;
    char ipAddress[NI_MAXHOST];
    unsigned scope=0;
    // walk over the list of all interface addresses
    getifaddrs(&addrs);
    for(struct ifaddrs *addr=addrs; addr; addr=addr->ifa_next){
        if (addr->ifa_addr && addr->ifa_addr->sa_family==AF_INET6){ // only
interested in ipv6 ones
            getnameinfo(addr->ifa_addr, sizeof(struct
sockaddr_in6), ipAddress, sizeof(ipAddress), NULL, 0, NI_NUMERICHOST);
            // result actually contains the interface name, so strip it
            for(int i=0; ipAddress[i]; i++){
                if(ipAddress[i]!='%'){
                    ipAddress[i]='\0';
                    break;
                }
            }
            // if the ip matches, convert the interface name to a scope index
            if(strcmp(ipAddress, ip)==0){
                scope=if_nametoindex(addr->ifa_name);
                break;
            }
        }
    }
    freeifaddrs(addrs);
    return scope;
}
```

answered Apr 17 '14 at 20:19



user3546716

51 1 1